

**IN THE CLAIMS:**

Please amend the claims as follows.

- B1
1. (Original) An instruction segment storing method, comprising:  
building an instruction segment according to Program Flow  
~~دetermine whether the instruction segment satisfies a filtering condition, and~~  
if the instruction segment satisfies the filtering condition, storing the instruction segment in a segment cache.
  2. (Currently Amended) The method of claim 1, wherein the filtering condition ~~may be met~~ only is met if all instructions in the instruction segment were assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.
  3. (Currently Amended) The method of claim 1, wherein the filtering condition ~~may be met~~ only is met if at least one instruction in the instruction segment was assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.
  4. (Currently Amended) The method of claim 1, wherein the filtering condition ~~may be met~~ only is met if a predetermined number of instructions in the instruction segment assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.
  5. (Currently Amended) The method of claim 1, wherein the filtering condition ~~may be met~~ only is met if an instruction of the segment by which the segment is to be indexed was assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.
  6. ~~دetermine whether the instruction segment satisfies a filtering condition, and~~  
(Currently Amended) An instruction segment storing method, comprising:  
~~building an instruction segment from fetched instructions,~~  
determining, from ~~location flags~~ identifying locations from which the associated with  
instructions in the instruction segment were fetched, whether the instruction segment satisfies a filtering condition, and  
if so, storing the instruction segment in a segment cache.

B1  
7. (Currently Amended) The method of claim 6, wherein the filtering condition ~~may be met~~ only is met if all instructions in the instruction segment were assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

8. (Currently Amended) The method of claim 6, wherein the filtering condition ~~may be met~~ only is met if at least one instruction in the instruction segment was assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

9. (Currently Amended) The method of claim 6, wherein the filtering condition ~~may be met~~ only is met if a predetermined number of instructions in the instruction segment were assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

10. (Currently Amended) The method of claim 6, wherein the filtering condition ~~may be met~~ only is met if an instruction of the segment by which the segment is to be indexed was assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

11. (Currently Amended) A front end system for a processing agent, comprising:  
an instruction cache system, and  
an instruction segment system, comprising:  
a segment cache, and  
a segment builder provided in communication with the instruction cache system,  
to store a new instruction segment in the segment cache after the instruction segment  
53 ~~has been built a predetermined plural number of times when a filtering condition is met.~~

12. (Currently Amended) The front end system of claim 11, further comprising a history map provided in communication with the segment builder to identify instruction segments that have been built previously but were discarded ~~when the filtering condition is met.~~

13. (Original) The front end system of claim 12, wherein the history map is a direct mapped cache.

14. (Original) The front end system of claim 12, wherein the history map is a set associative cache.

B1  
15. (Original) The front end system of claim 14, wherein the history map comprises a plurality of cache entries having a width corresponding to a width of a tag address of an instruction pointer in the system.

16. (Original) The front end system of claim 14, wherein the history map comprises a plurality of cache entries having a width corresponding to a width of a portion of a tag address of an instruction pointer in the system.

17. Canceled.

~~17~~  
17. (Currently Amended) A processing agent, comprising:  
a cache hierarchy, and  
a front end system comprising:

an instruction cache system in communication with the cache hierarchy, and  
an instruction segment system, comprising:

a segment cache, and

a segment builder provided in communication with the instruction cache system, to store a new instruction segment in the segment cache ~~after the instruction segment has been built a predetermined plural number of times when a filtering condition is met.~~

Sub C4  
~~18~~  
18. (Currently Amended) The processing agent of claim ~~18~~<sup>17</sup>, further comprising a history map provided in communication with the segment builder to identify instruction segments that have been built previously but were discarded ~~when the filtering condition is met.~~

20. Canceled.

~~19~~  
19. (Original) A computer system, comprising the processing agent of claim ~~18~~<sup>17</sup>, wherein the cache hierarchy includes an internal cache and a system memory.

~~20~~  
20. (Original) A computer system, comprising the processing agent of claim ~~18~~<sup>17</sup>, wherein the cache hierarchy includes an internal cache and an external cache.

~~21~~  
21. (Currently Amended) A method, comprising:  
building an instruction segment, *according to program flow*

B1. determining whether the instruction segment satisfies a filtering condition based on source locations of instructions within the instruction segment, and

storing the instruction segment in a segment cache unless the instruction segment does not satisfy the filtering condition.

<sup>23</sup>~~24~~. (Previously Presented) The method of claim <sup>21</sup>~~23~~, wherein the determining comprises determining whether all instructions in the instruction segment were assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

<sup>23</sup>~~25~~. (Previously Presented) The method of claim <sup>21</sup>~~23~~, wherein the determining comprises determining whether at least one instruction in the instruction segment assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

<sup>24</sup>~~26~~. (Previously Presented) The method of claim <sup>21</sup>~~23~~, wherein the determining comprises determining whether a predetermined number of instructions in the instruction segment assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

<sup>25</sup>~~27~~. (Previously Presented) The method of claim <sup>21</sup>~~23~~, wherein the determining comprises determining whether an instruction of the segment by which the segment is to be indexed was assembled into the instruction segment from an instruction cache of a front-end processing system in a processor.

28. Canceled.

<sup>24</sup>~~28~~. (Previously Presented) The method of claim <sup>21</sup>~~23~~, wherein the determining comprises comparing location flags identifying locations from which instructions within the segment were retrieved to a predetermined filtering condition.

<sup>27</sup>~~30~~. (Currently Amended) A front end system for a processing agent, comprising:  
an instruction cache system, and  
an instruction segment system, comprising:

Sub  
CS ~~a segment builder to build instruction segments from instructions retrieved from the instruction cache system,~~

B1 a segment cache to store instruction segments ~~unless after the respective instruction segments have been built at least twice~~~~fail a filtering condition~~.

<sup>28</sup>~~31~~. (Currently Amended) The front end system of claim <sup>27</sup>~~30~~, further comprising a history map provided in communication with the segment builder to identify instruction segments that have been built but not stored in the segment cache~~when the filtering condition is met~~.

<sup>29</sup>~~32~~. (Previously Presented) The front end system of claim <sup>28</sup>~~31~~, wherein the history map is a direct mapped cache.

<sup>30</sup>~~33~~. (Previously Presented) The front end system of claim <sup>28</sup>~~31~~, wherein the history map is a set associative cache.

<sup>31</sup>~~34~~. (Previously Presented) The front end system of claim <sup>30</sup>~~33~~, wherein the history map comprises a plurality of cache entries having a width corresponding to a width of a tag address of an instruction pointer in the system.

<sup>32</sup>~~35~~. (Previously Presented) The front end system of claim <sup>30</sup>~~33~~, wherein the history map comprises a plurality of cache entries having a width corresponding to a width of a portion of a tag address of an instruction pointer in the system.

[ <sup>36</sup> 36. Canceled.

[ Please add the following new claims: ]

<sup>33</sup>~~37~~. (New) An instruction segment storing method, comprising:  
building instruction segments as instructions are executed,  
determining whether a new instruction segment has been built multiple times, and  
if so, storing the new instruction segment in a segment cache,  
otherwise, discarding the new instruction segment.

<sup>34</sup>~~38~~. (New) The method of claim <sup>33</sup>~~37~~, further comprising storing identifiers of discarded instruction segments for comparison against identifiers of newly built instruction segments.

<sup>35</sup>~~39~~. (New) The method of claim <sup>33</sup>~~37~~, wherein the instruction segment is a basic block.

B1 <sup>34</sup>~~40~~. (New) The method of claim <sup>33</sup>~~37~~, wherein the instruction segment is a trace and is indexed within the segment cache by a first instruction therein.

<sup>31</sup>~~41~~. (New) The method of claim <sup>33</sup>~~37~~, wherein the instruction segment is an extended block and is indexed within the segment cache by a last instruction therein.